In [1]:

```
# importing various libraries which are used
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import preprocessing as p
from scipy.stats import f
from sklearn.metrics import mean_squared_error, mean_absolute_error ,r2_score
```

In [2]:

```
# taking the csv file as input to create the model
file=input("csv file_name : ")
```

```
csv file_name : 505.csv
```

In [3]:

```
# creating a pandas dataframe using the values obtained in the csv
df=pd.read_csv(file)
```

In [4]:

```
# printing the head of the data frame to get the gist of the values
print("\n data frame head :- \n",df.head())
```

```
 data frame head :-
        time  cpu-cycles  instructions  l1d.replacement  icache_64b.if
tag_miss  \
0  0.100147   456035951     806165718         38117417
446879
1  0.200389   459116285     574087409         25271248
17055
2  0.300591   458958052     498238976         18428549
12684
3  0.400808   459248163     473123154         16188576
11227
4  0.501016   458795107     456722529         15050320
16811

   l2_rqsts.all_demand_miss  longest_lat_cache.miss  \
0                  11932692                 5708352
1                   9648088                 3316697
2                   6552234                 2506574
3                   5359297                 1891785
4                   4771375                 1874954

   br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                     216907802                         558
1                     144096170                          21
2                     124549108                          15
3                     117789280                          14
4                     113905609                          28

   itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                         864                            18983
1                         156                            32213
2                         140                            25784
3                         107                            21720
4                         184                            18725

   dtlb_store_misses.walk_completed  branch-misses
0                             25383        2187370
1                               295        6551907
2                               220        7746323
3                               194        8179764
4                               249        8328905
```

In [5]:

```
# creating a new col in our data frame which consists of the CPI per tuple
df[['CPI']]=df[['cpu-cycles']].div(df['instructions'], axis=0)
print(df)
```

```
            time  cpu-cycles  instructions  l1d.replacement  \
0       0.100147   456035951     806165718         38117417
1       0.200389   459116285     574087409         25271248
2       0.300591   458958052     498238976         18428549
3       0.400808   459248163     473123154         16188576
4       0.501016   458795107     456722529         15050320
...          ...         ...           ...              ...
2438  244.438602   458386917     538151439         51057807
2439  244.538809   458870554     455912689        106772858
2440  244.639004   459206801     464983529        107429771
2441  244.739162   459126745     464354921        107161921
2442  244.796317   261468012     425005741         33487994

      icache_64b.iftag_miss  l2_rqsts.all_demand_miss  longest_lat_cac
he.miss  \
0                    446879                  11932692
5708352
1                     17055                   9648088
3316697
2                     12684                   6552234
2506574
3                     11227                   5359297
1891785
4                     16811                   4771375
1874954
...                     ...                       ...
...
2438                  13291                   9658999
7744748
2439                   9278                   9970511
32814
2440                  10588                   9900047
21865
2441                   9925                   9945010
18355
2442                  14286                   4806312
1210327

      br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                        216907802                         558
1                        144096170                          21
2                        124549108                          15
3                        117789280                          14
4                        113905609                          28
...                            ...                         ...
2438                     120147953                          13
2439                      90987832                          18
2440                      92030875                          19
2441                      92599514                          33
2442                      60864405                          78

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                            864                            18983
1                            156                            32213
2                            140                            25784
3                            107                            21720
4                            184                            18725
...                          ...                              ...
2438                         114                            45291
2439                          17                              286
2440                           6                              754
```

```
2441                            10                                 116
2442                           218                                5362

      dtlb_store_misses.walk_completed  branch-misses       CPI
0                                25383        2187370  0.565685
1                                  295        6551907  0.799732
2                                  220        7746323  0.921160
3                                  194        8179764  0.970674
4                                  249        8328905  1.004538
...                                ...            ...       ...
2438                            262281        1650317  0.851781
2439                               164          92906  1.006488
2440                               166          53614  0.987576
2441                               149          43149  0.988741
2442                              8318          95543  0.615211

[2443 rows x 14 columns]
```

In [6]:

```
# dividing all the values by instruction so that we get values in each coloumn per i
df[['l1d.replacement','icache_64b.iftag_miss','l2_rqsts.all_demand_miss','longest_la
print(df)
```

```
            time  cpu-cycles  instructions  l1d.replacement  \
0       0.100147   456035951     806165718         0.047282
1       0.200389   459116285     574087409         0.044020
2       0.300591   458958052     498238976         0.036987
3       0.400808   459248163     473123154         0.034216
4       0.501016   458795107     456722529         0.032953
...          ...         ...           ...              ...
2438  244.438602   458386917     538151439         0.094876
2439  244.538809   458870554     455912689         0.234196
2440  244.639004   459206801     464983529         0.231040
2441  244.739162   459126745     464354921         0.230776
2442  244.796317   261468012     425005741         0.078794

      icache_64b.iftag_miss  l2_rqsts.all_demand_miss  longest_lat_cac
he.miss  \
0                  0.000554                  0.014802
0.007081
1                  0.000030                  0.016806
0.005777
2                  0.000025                  0.013151
0.005031
3                  0.000024                  0.011327
0.003999
4                  0.000037                  0.010447
0.004105
...                     ...                       ...
...
2438               0.000025                  0.017948
0.014391
2439               0.000020                  0.021869
0.000072
2440               0.000023                  0.021291
0.000047
2441               0.000021                  0.021417
0.000040
2442               0.000034                  0.011309
0.002848

      br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                         0.269061                6.921654e-07
1                         0.251000                3.657980e-08
2                         0.249979                3.010603e-08
3                         0.248961                2.959060e-08
4                         0.249398                6.130637e-08
...                            ...                         ...
2438                      0.223260                2.415677e-08
2439                      0.199573                3.948124e-08
2440                      0.197923                4.086166e-08
2441                      0.199415                7.106633e-08
2442                      0.143208                1.835269e-07

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                   1.071740e-06                     2.354727e-05
1                   2.717356e-07                     5.611166e-05
2                   2.809897e-07                     5.175027e-05
3                   2.261568e-07                     4.590771e-05
4                   4.028704e-07                     4.099863e-05
...                          ...                              ...
2438                2.118363e-07                     8.416032e-05
2439                3.728784e-08                     6.273131e-07
2440                1.290368e-08                     1.621563e-06
```

```
2441                 2.153525e-08                        2.498089e-07
2442                 5.129342e-07                        1.261630e-05

      dtlb_store_misses.walk_completed  branch-misses       CPI
0                           3.148608e-05       0.002713  0.565685
1                           5.138590e-07       0.011413  0.799732
2                           4.415552e-07       0.015547  0.921160
3                           4.100412e-07       0.017289  0.970674
4                           5.451888e-07       0.018236  1.004538
...                                  ...            ...       ...
2438                        4.873740e-04       0.003067  0.851781
2439                        3.597180e-07       0.000204  1.006488
2440                        3.570019e-07       0.000115  0.987576
2441                        3.208752e-07       0.000093  0.988741
2442                        1.957150e-05       0.000225  0.615211

[2443 rows x 14 columns]
```

In [7]:

```
# droping values such as time , instructions , cpu-cycles and br_inst_retired.all_br
df= df.drop(['time'], axis=1)
df= df.drop(['instructions'], axis=1)
df= df.drop(['cpu-cycles'], axis=1)
df= df.drop(['br_inst_retired.all_branches'], axis=1)
```

In [8]:

```
# assigning y as the CPI and then droping it from the dataframe
y=df['CPI']
df= df.drop(['CPI'], axis=1)
print("y values :- \n",y)
```

```
y values :-
 0       0.565685
1       0.799732
2       0.921160
3       0.970674
4       1.004538
          ...
2438    0.851781
2439    1.006488
2440    0.987576
2441    0.988741
2442    0.615211
Name: CPI, Length: 2443, dtype: float64
```

In [9]:

```
# assigning x as the dataframe
x=df
print("x values :- \n",x)
```

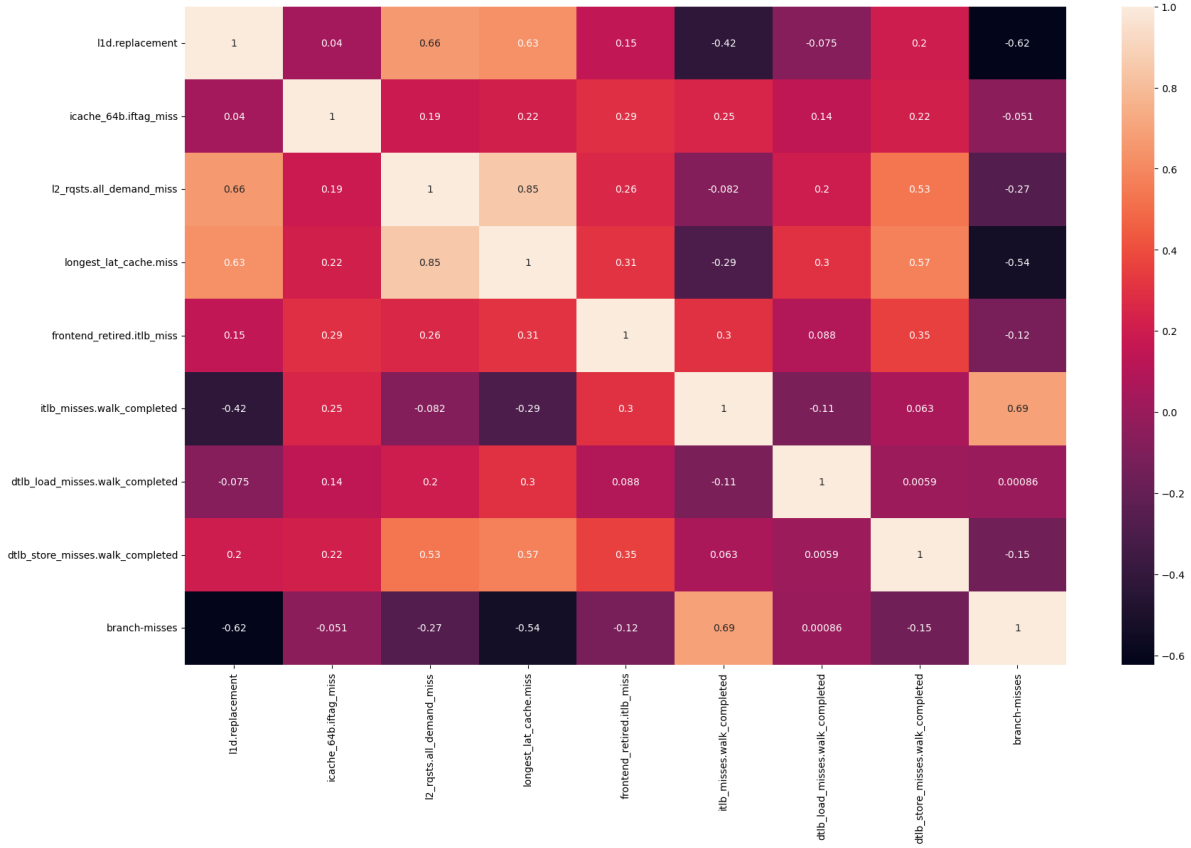
```
x values :-
       l1d.replacement  icache_64b.iftag_miss  l2_rqsts.all_demand_mis
s  \
0             0.047282                0.000554                  0.014802
1             0.044020                0.000030                  0.016806
2             0.036987                0.000025                  0.013151
3             0.034216                0.000024                  0.011327
4             0.032953                0.000037                  0.010447
...                ...                     ...                       ...
2438          0.094876                0.000025                  0.017948
2439          0.234196                0.000020                  0.021869
2440          0.231040                0.000023                  0.021291
2441          0.230776                0.000021                  0.021417
2442          0.078794                0.000034                  0.011309

      longest_lat_cache.miss  frontend_retired.itlb_miss  \
0                   0.007081                6.921654e-07
1                   0.005777                3.657980e-08
2                   0.005031                3.010603e-08
3                   0.003999                2.959060e-08
4                   0.004105                6.130637e-08
...                      ...                         ...
2438                0.014391                2.415677e-08
2439                0.000072                3.948124e-08
2440                0.000047                4.086166e-08
2441                0.000040                7.106633e-08
2442                0.002848                1.835269e-07

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                   1.071740e-06                     2.354727e-05
1                   2.717356e-07                     5.611166e-05
2                   2.809897e-07                     5.175027e-05
3                   2.261568e-07                     4.590771e-05
4                   4.028704e-07                     4.099863e-05
...                          ...                              ...
2438                2.118363e-07                     8.416032e-05
2439                3.728784e-08                     6.273131e-07
2440                1.290368e-08                     1.621563e-06
2441                2.153525e-08                     2.498089e-07
2442                5.129342e-07                     1.261630e-05

      dtlb_store_misses.walk_completed  branch-misses
0                         3.148608e-05       0.002713
1                         5.138590e-07       0.011413
2                         4.415552e-07       0.015547
3                         4.100412e-07       0.017289
4                         5.451888e-07       0.018236
...                                ...            ...
2438                      4.873740e-04       0.003067
2439                      3.597180e-07       0.000204
2440                      3.570019e-07       0.000115
2441                      3.208752e-07       0.000093
2442                      1.957150e-05       0.000225

[2443 rows x 9 columns]
```

In [10]:

```
# creating a heatmap of the correlation matrix
fig,axis = plt.subplots(figsize = (20,12))
sns.heatmap(x.corr(),annot=True)
```

Out[10]:

```
<AxesSubplot:>
```



In [11]:

```
# dividing the data set into test and train set in a 20:80 ration with a random stat
# the model trains on a particular set of values on every execution
X_train, X_test, y_train, y_test = train_test_split(
    x, y, test_size=.20,random_state=55)
```

In [12]:

```
# using MinMax Scaler to scale the data within the given range of 0 to 1 such that
#shape of the original distribution is same after transformation
mms = p.MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.transform(X_test)
```

In [13]:

```
print("X_train :-\n",X_train)
```

```
X_train :-
 [[4.48686499e-01 6.04562768e-02 2.42360085e-01 ... 5.77516453e-02
  4.68332793e-03 2.50920899e-02]
 [2.82759163e-01 2.25465833e-02 3.65142738e-01 ... 1.31915228e-02
  2.23266507e-04 4.05205177e-01]
 [1.36162068e-01 2.73845868e-02 1.70185727e-01 ... 1.00986224e-02
  1.05674620e-04 5.84538884e-01]
 ...
 [5.39142555e-02 2.78262051e-02 1.03307290e-01 ... 7.64577382e-03
  5.98342124e-05 8.19486981e-01]
 [5.53081686e-01 1.14115137e-01 8.46427729e-01 ... 8.57663777e-02
  8.58044342e-01 3.36972321e-01]
 [8.86843137e-02 2.78742179e-02 1.67017405e-01 ... 8.69868567e-03
  8.73222016e-05 7.97414406e-01]]
```

In [14]:

```
# mean of all the columns of the training set
df2 = X_train.mean(axis=0)
print(df2)
```

```
[0.16253495 0.02822497 0.17020972 0.12551652 0.04681522 0.2143658
 0.0634587  0.00975975 0.5588797 ]
```

In [15]:

```
# creating a linear regression model using sklearn.linear_model
model = LinearRegression(positive=True)
model.fit(X_train,y_train)
```

Out[15]:

```
LinearRegression(positive=True)
```

In [16]:

```
# finding the coefficients given by our model
c=model.coef_
print("\nCoefficients :- \n",c)
```

```
Coefficients :-
 [0.08780327 0.35854264 0.         1.68344848 0.18203317 0.
 0.40899258 3.10859763 0.66730589]
```

In [17]:

```
# model intercept i.e. the " Base CPI "
i=model.intercept_
print("\nBase CPI : ",i)
```

```
Base CPI :  0.5305186711143667
```

In [18]:

```
# making the predictions using our model on the test set
predictions = model.predict(X_test)
```

In [19]:

```
# Actual CPI
ACPI = y_test.mean()
print("\n Actual CPI : ",ACPI)
```

```
 Actual CPI :  1.2444685430625668
```

In [20]:

```
# Predicted CPI
PCPI = predictions.mean()
print("\n Predicted CPI : ",PCPI)
```

```
 Predicted CPI :  1.250469995412093
```

In [21]:

```
# Finding out RMSE , R^2 , adjusted R^2 using our predictions and test set
RMSE = mean_squared_error(y_test, predictions)
print("\n RMSE : ",RMSE)

R2 = r2_score(y_test, predictions)
print("\n R^2 : ",r2_score(y_test, predictions))

adjusted_r2 = 1 - ( 1-model.score(X_test,y_test) ) * ( len(y_test) - 1 ) / ( len(y_t
print("\n adjusted R^2 : ",adjusted_r2)
```

```
 RMSE :  0.012633184499023625

 R^2 :  0.954627362313952

 adjusted R^2 :  0.9537748492885356
```

In [22]:

```
# finding absolute error and accuracy on test set
err = mean_absolute_error(y_test, predictions)
print ( "\n Test error is :" , err *100 , "% " )
print ( "\n Test Accuracy is :" , (1- err) *100 , "%" )
```

```
 Test error is : 7.448294524082144 %

 Test Accuracy is : 92.55170547591786 %
```

In [23]:

```
# F-statistic value which should be > 2.5 and p-value which should be < 0.05
F = (R2/(1-R2))*((X_test.shape[0]-1-X_test.shape[1])/X_test.shape[1])
print("\n F-statistic : ",F)

p = 1-f.cdf(F,X_test.shape[1],(X_test.shape[0]-1-X_test.shape[1]))
print("\n p-value : ",p)
```

```
 F-statistic :  1119.7803832354698

 p-value :  1.1102230246251565e-16
```

In [24]:

```
#no of coefficients
X_test.shape[1]
```

Out[24]:

```
9
```

In [25]:

```
# no of tuples in the test set
X_test.shape[0]
```

Out[25]:

```
489
```

In [26]:

```
# finding the residual for our test set
residuals = y_test - predictions
print("\n Residual :- \n ",residuals)
```

```
 Residual :-
  103     0.013822
893     0.037199
2234   -0.146452
1111   -0.063420
517    -0.191404
          ...
544    -0.122094
864     0.156913
1302   -0.146050
1326    0.067084
779    -0.102042
Name: CPI, Length: 489, dtype: float64
```

In [27]:

```
# residual graph
data = {
    'predicted': [i for i in predictions],
    'residuals': [i for i in residuals]
}

dfr = pd.DataFrame(data)
sns.scatterplot(data=dfr, x="predicted", y="residuals")
```

Out[27]:

```
<AxesSubplot:xlabel='predicted', ylabel='residuals'>
```